Attorney's Docket No.: 06666/076001 / USC-2892

APPLICATION

FOR

UNITED STATES LETTERS PATENT

TITLE: THREE-DIMENSIONAL MODELING AND BASED ON PHOTOGRAPHIC IMAGES

APPLICANT: QIAN CHEN AND GERARD MEDIONI

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EL584937407US

I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail Post Office to Addressee with sufficient postage on the date indicated below and is addressed to the Commissioner for Patents, Washington, D.C. 20231.

May 3, 2001
Date of Deposit

Signature

Gil Vargas
Typed or Printed Name of Person Signing Certificate

THREE-DIMENSIONAL MODELING AND BASED ON PHOTOGRAPHIC IMAGES

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of the U.S. Provisional Application No. 60/201,585, filed 05/03/00, and U.S. Provisional Application No. 60/213,393, filed 06/23/00.

BACKGROUND

[0002] Many different applications exist for three-dimensional imaging. While many current image-viewing media, such as display screens and photographs, display only in two dimensions, the information from the third dimension may still be useful, even in such two-dimensional displays.

[0003] For example, teleconferencing may be used to allow several geographically separate participants to be brought into a single virtual environment. Three dimensional information may be used in such teleconferencing, to provide realism and an ability to modify the displayed information, to accommodate facial movement.

[0004] Facial orientation and expression may be used to drive models over the network to produce and enhance the realism.

[0005] Three-dimensional information may also be usable over a network, using, for example, the concept of cyber touch. Those people browsing the web page of a certain server such as a museum may be allowed to touch certain objects using a haptic device. One such device is available at http://digimuse.usc.edu/IAM.htm.

[0006] Work along this line has been carried out under the names aerial triangulation, and binocular stereo.

[0007] Three-dimensional models may be obtained using a laser scanner. Other techniques are also known for obtaining the three-dimensional models. Practical limitations, however, such as cost, complexity, and delays may hamper obtaining an accurate three-dimensional model.

[0008] If two cameras are completely calibrated, then obtaining a full 3D model from 2D information is known. See the book "Multiple View Geometry in Computer Vision", Richard Hartley and Andrew Zisserman, Cambridge University Press, June 2000. Calibration of cameras includes internal calibration and external calibration. Internal calibration refers to the characteristic parameters of the camera such as focal length, optical center, distortion, skew,.... External parameters which describe the relative position and orientation of the cameras with respect to each other. It is known to go to 3D from a

disparity map if cameras are both internally and externally calibrated.

[0009] Internal calibration of cameras is a well understood problem in the literature, with packages freely available to perform this step. Intel's OpenCV Library at http://www.intel.com/research/mrl/research/opencv/, for example, can be used. These techniques such as these may be used to internally calibrate the cameras offline. However, the present system does not require calibration.

SUMMARY

[0010] The present application teaches a technique of processing two-dimensional images such as photographs to obtain three-dimensional information from the two-dimensional photographs. Different aspects of the invention describe the ways in which multiple images are processed to obtain the three-dimensional information therefrom.

[0011] According to one aspect, the images are modified in a way that avoids the necessity to calibrate among the cameras.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] These and other aspects will now be described in detail with reference to the accompanying drawings, wherein:

[0013] Figure 1 shows a system flowchart;

[0014] Figure 2 shows a hardware block diagram;

[0015] Figure 3a and 3b show illustrations of epipolar geometry;

[0016] Figure 4a-4c show operations on images;

[0017] Figure 5 shows a flowchart of operation.

DETAILED DESCRIPTION

[0018] The binocular stereo technique used according to the present system may find three-dimensional information from two or more pictures taken from close locations. This system may find image portions, e.g. pixels, between the images that correspond. Rays from the two pixels are used to form a simulated three-dimensional location of a point on the 3-D item. All recovered three-dimensional points may then be connected to form a polygonal mesh representing aspects of the 3D shape of the object. The three-dimensional object may create photorealistic images at arbitrary viewpoints. These images may be useful for telepresence, in which several geographically separated participants may be brought into one virtual environment. Three-dimensional information may allow production of face models at remote sites, for example may be driven at arbitrary orientations.

[0019] The inventors have recognized a number of problems which exist in forming three-dimensional information based on

similarity peaks between the different information. In addition, previous systems have caused false matches, thereby showing an irregularity in the final model. Previous systems have also required that either the cameras be calibrated, or certain extra mathematical steps be followed to ensure that the uncalibrated cameras do not cause false results.

[0020] One aspect of the present system uses a semi automatic approach to address limitations in the prior art. Computer vision techniques may be used with a semi-automatic approach. Manual techniques may be used for initialization and to determine parts of the reconstruction that are acceptable and other parts that are not.

[0021] The basic operation is shown in the overall flowchart of Figure 1. The Figure 1 flowchart may be carried out on any machine which is capable of processing information and obtaining vision type information. An exemplary hardware system is shown in Figure 2. The Figure 2 embodiment shows use of two cameras 200, 205. The two cameras may be Fuji model DS-300 cameras. A synchronizing device 210 provides synchronization such that the cameras are actuated at substantially the same time. However, synchronization need not be used if the subject stays relatively still.

[0022] In another embodiment, the same camera is used to obtain two images that are offset from one another by some small amount, e.g., less than 15 degrees.

[0023] The camera outputs may be colored images with 640 by 480 resolution. Color is not in fact necessary, and may be used only in the texture mapping part of the applications. The images are input into the system as a pair of stereo images 100, 105. The stereo images are preferably images of the same scene from slightly different angles.

[0024] Alternate embodiments may use other digital cameras, such as cameras connected by USB or Firewire, and can include analog or video cameras. Other resolutions, such as 320x240 and 1600x1200 may also be used.

[0025] Manual selection is used to allow the user to select specified corresponding points in the two images at 110. The locations of those manually-selected corresponding points are then refined using automatic methods. Moreover, the system may reject selected points, if those selected points do not appropriately match, at 115.

[0026] Alternatively, the system may use a totally automatic system with feature points and robust matching, as described by Zhang et al, or Medioni-Tang [C.-K. Tang, G. Medioni and M.-S. Lee, ``Epipolar Geometry Estimation by Tensor Voting in 8D,''

in Proc. IEEE International Conference on Computer Vision (ICCV), Corfu, Greece, September 1999].

[0027] At 120, the system computes the " fundamental matrix" based on this manual input. The fundamental matrix is well known in the art as a Rank 2, 3x3 matrix that describes information about images in epipolar geometry.

[0028] An alternative may allow automatic establishing of correspondence if high entropy parts are included in the image. For example, if the image has high-intensity curvature points such as eye corners of a human face, then these points may be used to automatically establish correspondence.

[0029] The fundamental matrix at 120 may be used to automatically align the two images to a common image plane. The aligned images are automatically matched.

[0030] 125 represents carrying out image rectification. In general, the two cameras 200, 205 that are used to generate the stereo images 100, 105 are not parallel. Rectification is used to align the two image planes from the two cameras 200, 205. This effectively changes the numerical representation of the two images so that the two stereo images become coplanar and have scan lines that are horizontally parallel.

[0031] The system used according to the present technique may rely on epipolar geometry, as described herein. This geometry

is between two views of the same scene and is algebraically described by the fundamental matrix.

[0032] The image space is treated as a two-dimensional projective space P^2 which has certain properties. In this space, points and lines become dual entities. For any projective result established using these points and lines, a symmetrical result holds. In this result, the roles of the lines and points are interchanged. Points may become lines, and lines may become points in this space.

[0033] Graphically, epipolar geometry is depicted in Figure 3B where P, P' are 3-D scene points; $\boldsymbol{p}_1$, $\boldsymbol{p}_2$ are images of P. O_1, O_2 are camera projection centers. The line O_1O_2 is called the *baseline*. Notice that the two triangles ΔO_1O_2P and $\Delta O_1O_2P'$ are coming from a pencil-of-planes which is projected to the pencil-of-lines in the image planes. The latter (e.g. $\boldsymbol{l}_1$ and $\boldsymbol{l}_2$) form *epipolar lines*. The intersection of each pencil-of-lines is called the *epipole* ($\boldsymbol{o}_1$, $\boldsymbol{o}_2$). An epipole has many interesting characteristics. It is the intersection of all the epipolar lines, and it is also the intersection of the baseline with the image plane. It is also the projection of a camera projection center on the counterpart image plane. It is observable from that if an image plane is parallel to the baseline, then its epipole is at infinity and then all epipolar lines on that image plane become parallel.

[0034] Algebraically, epipolar geometry is described by the following equation:

$$p_2^T F p_1 = 0$$

where F is the fundamental matrix, e.g. a 3x3 rank 2 matrix and $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ are the 3-tuple homogeneous coordinates of corresponding pixel points. $\boldsymbol{p}_1$ is located on the epipolar line defined by $\boldsymbol{p}_2^T F$. The relationship is symmetric: $\boldsymbol{p}_2$ is on the line defined by $\boldsymbol{p}_1^T F^T$. Since $\boldsymbol{o}_1^T F^T \boldsymbol{p}_2 = 0$ for any $\boldsymbol{p}_2$, $F\boldsymbol{o}_1 = 0$. Thus $\boldsymbol{o}_1$ is the null vector of F which reflects the fact that F is of rank 2. Similarly, $\boldsymbol{o}_2$ is the null vector of F^T. It is observable from that if an image plane is parallel to the baseline, then its epipole is at infinity and then all epipolar lines on that image plane become parallel.

A rectification transformation over an image plane may be represented by a 3x3 matrix with eight independent coefficients ignoring a scale factor. This may require at least 4 correspondences to compute a rectification transformation.

[0035] A transformation may be represented by a 3x3 matrix with eight independent coefficients and a scale factor. This may produce 4 correspondences.

[0036] Figure 3A illustrates the rectification technique in this epipolar geometry. L1, R1 forms a first pair of epipolar lines with L2, R2 being the second pair of epipolar lines. Note

that these lines, however, are not properly aligned, and cannot be aligned as shown. To align, a new line v1 is defined, which passes through the average of the y coordinates of the start and end points corresponding to beginning and end of L1 and beginning and end of R1. Another line v2 is similarly formed from L2, R2. These lines, v1 and v2 are aligned. Accordingly, this rectification transformation may map the non-aligning lines, L1-R1, L2-R2 to the aligned lines v1, v2. From the intersections of these lines at the vertical image edges, a rectification matrix can be computed for each image.

[0037] The software provided by "Zhang" is used for the fundamental matrix computation. See http://wwe-sop.inrld.fr/robotvis/demo/f-http/html/. Below, a brief explanation of why this technique works is provided.

[0038] First, recall from the above explanation section that a cross-ration between the two pencil-of-lines in the two image planes is unchanged, and that cross-ratio is invariant to homography.

[0039] Second, within each pencil, the line that is "at infinity" is the one that passes the image origin; this line possesses the canonical form [*a*, *b*, 0].

[0040] Third, after the rectification, all three bases are aligned - two of them form the top and bottom edges of the "trapeze", the special one mentioned above is mapped to the X-

axis. This fact plus the invariant property of cross-ratio makes the alignment of the corresponding epipolar lines, or scanlines after rectification, useful.

[0041] Figures 4A-4C shows a rectification process which operates to superimpose the epipolar lines. First matching points are automatically selected by Zhang's software. These are shown as points in Figure 4A. In Figure 4B, lines before extraction are shown. Consider for example line 400 and 401. Both of these lines pass through the same point through the user's eye in the two different images 405, 410. However, the lines 400, 401 do not line up in the two images. Similarly, other lines which pass through the same corresponding points in other images do not line up. Consider, for example, line 416 which passes through the right eye quarter in both images 405 and 410. This does not line up with the line 417 in image 405.

[0042] In order to align these images, the rectification transformation is carried out to produce the images shown in Figure 4C. In these images, each of the lines line up. Specifically, the lines 416 lines up with 417, and 400 lines up with 401. Each of the other lines also aligns. Importantly, this all can be done based on information in the fundamental matrix. This can be recovered from the eight pairs of point correspondences as described above. This can be recovered from the eight pairs of point correspondences as described above. The

epipolar geometry is an output of Zhang's software. The rectification transformation matrix is calculated by the present system, in contrast, using only four pairs of correspondences.

[0043] By using this transformation, therefore, full camera calibration may be avoided, and instead the thus-obtained information can be used.

[0044] At 130, the aligned images are matched. Image matching has been typically formulated as a search optimization problem based on local similarity measurements. Global constraints may be enforced to resolve ambiguities, such as multiple matches. The correspondence information may be represented as disparity, *d* which may be conceptualized as the difference between the axial coordinates of the two matching pixels.

[0045] The disparity coordinate *d* may be a function of the pixel coordinates *(u,v)* of the images. Accordingly, d*(u,v)* may define a piecewise continuous surface over the domain of the image plane. This surface *d(u,v)* is referred to as this disparity surface.

[0046] Image matching can therefore be thought of as location of the disparity surface in abstract three-dimensional space. The output is the disparity map recording the disparity value *d* as a function of the pixels *u,v*.

[0047] The image matching in 130 embeds the disparity surface into a volume. Each voxel in the volume has a value proportional to the probability of the volume being on the disparity surface. Hence, the image matching is carried out by extremal surface extraction. Discrete surface patches may be found using volume rendering techniques.

[0048] Mathematically, image matching can be encoded as the correspondence information by a function $d(u,v)$ defined over a first image plane (denoted as I_1) such that (u,v) and $(u+d(u,v), v)$ become a pair of corresponding pixels. Geometrically, $d(u,v)$ defines the disparity surface. Assuming corresponding pixels have similar intensities (color), and letting Φ denote a similarity function such that larger values mean more similar pixels, matching can be formulated as a variational problem:

$$D(u,v) = \max_{d(u,v)} \left(\iint_{I_1} \Phi(u,v,d(u,v)) du dv \right). \quad (1)$$

[0049] One simple solution to (1) is to sample over all possible values of u, v, and d, followed by an exhaustive search in the resulting volume. However, it is desirable to do this "efficiently". There are two issues: one is efficiency—how to i.e. to perform the search in a time-efficient way; and robustly i.e. to avoid local extrema.

[0050] In the disclosed technique $d(u,v)$ is treated geometrically as a *surface* in a volume instead of an algebraic

function. The surface is extracted by propagating from seed voxels which have relatively high probability of being correct matches.

[0051] A normalized cross-correlation over a window as the similarity function:

$$\Phi(u,v,d) = \frac{Cov(W_l(u,v), W_r(u+d,v))}{Std(W_l(u,v)) \cdot Std(W_r(u+d,v))} \quad (2)$$

[0052] where W_l and W_r are the intensity vectors of the left and right windows of size W centered at (u,v) and $(u+d,v)$ respectively, d is the disparity, "*Cov*" stands for covariance and "*Std*" for standard deviation. The width and height of the (left) image together with the range of d form the *u-v-d volume*. The range of Φ is [0 -> 1]. When Φ is close to 1, the two pixels are well correlated, hence have high probability of being a match. When Φ is close to -1, that probability is low. In implementation, a threshold needs to be set. We discuss how to choose its value in the next subsection.

[0053] The fact that Φ is a local maximum when (u,v,d) is a correct match means that the disparity surface is composed of voxels with peak correlation values. Matching two images is therefore equivalent to extracting the maximal surface from the volume. Since the *u-v-d* volume may be very noisy, simply applying the "Marching Cubes" algorithm might easily fall into the trap of local maxima. A special propagation technique is

used along with the *disparity gradient limit* which states that $|\Delta d|/|\Delta u|<1$. Use of this constraint in the scanline direction is equivalent to the *ordering constraint* often used in scanline-based algorithms (e.g. by Cox *et al.*). Using it in the direction perpendicular to the scan lines enforces smoothness across scan lines, which is only partially enforced in inter-scanline based algorithms such as the one presented by Ohta and Kanade.

[0054] The output from this matching algorithm is the disparity map which corresponds to the voxels that comprise the disparity surface. As can be appreciated this is different from volume rendering, or other matching methods that model the disparity surface as a continuous function. The technique is shown in the flowchart of Figure 5.

[0055] First, at 500, a seed voxel is selected.

[0056] A voxel (u,v,d) is a seed voxel if,

[0057] it is unique – meaning for the pixel (u,v), there is only one local maximum at d along the scanline v, and

[0058] $\Phi(u,v,d)$ is greater than a threshold $t1$.

[0059] A seed should reside on the disparity surface. Otherwise, the true surface point (u,v,d'), for which $d'\neq d$, would be a second local maximum.

[0060] To find seeds, the image is divided into a number of parts or "buckets" at 502. Inside each bucket, pixels are checked randomly at 504 until either one seed is found, or all

pixels have been searched without success. During the search, the voxel values may be cached to save computation time for subsequently operating the next step. The value of $t1$ determines the confidence of the seed points. It may be set close to 1. In specific experiments, we start from 0.995 trying to find at least 10 seeds at 506. If too few seeds are found, the value is decreased. In all the examples tried so far, we have found the range of $t1$ to be between 0.993 and 0.996; more generally, the t1 should be greater than 0.9, even greater than 0.99.

[0061] At 510, surface tracing is carried out at 510.

[0062] The disparity surface may be traced simultaneously from all seed voxels, by following the local maximal voxels whose correlation values are greater than a second threshold $t2$. The $|\Delta d|/|\Delta u|<1$ constraint determines that when moving to a neighboring voxel, only those at d, d-1, d+1 need to be checked. Initially, the seed voxels may be in a first in-first out (FIFO) queue at 512. After tracing starts, the head of the queue is exposed every time, and the 4-neighbors of the corresponding pixel are checked at 514. Border pixels need special treatment. When two surface fronts meet, the one with the greater correlation value prevails. If any new voxels are generated, they are pushed to the end of the queue. This process continues at 516 until the queue becomes empty.

[0063] To enforce smoothness, the voxel (u', v', d) may be assigned higher priority than (u', v', d-1) and (u', v', d+1). To obtain sub-pixel accuracy, a quadratic function is fitted at (u', v', d'-1), (u', v', d'), and (u', v', d'+1) where (u', v', d') is the newly-generated voxel. $t2$ determines the probability that the current voxel is on the same surface that is being traced; however the value of $t2$ may not be critical. In all the examples tried so far, the value 0.6 is used. Exemplary pseudo code of the tracing algorithm is given in Table 1.

Algorithm 1. *Disparity Surface Tracing*

```
Initialize Q with the seed voxels;
While (not empty Q)
{
        Set (u, v, d) = pop Q;
        For each 4-neighbor of (u, v)
        {
                Call it (u', v');
                Choose among (u', v', d-1), (u', v', d), (u', v', d+1)
                        the one with the max correlation value and call it (u', v', d');
                if (u',v') already has a disparity d''
                        disparity(u', v') = Φ( u', v', d') > Φ (u', v', d'') ? d' : d'';
                else if Φ(u', v', d') > t2
                {
                        disparity(u', v') = d';
                        push (u', v', d') to the end of Q;
                }
        }
}
```

Table 1 Pseudo code of the tracing algorithm

[0064] The worst case complexity of the seed selection part is bounded by $\mathbf{O}(WHDW)$ where W and H are respectively the width and height of the image, D is the range of the disparity, and W is the size of the correlation window. The tracing part is bounded by $\mathbf{O}(WHW)$. Since some voxels have already been computed during initial seed voxel determination the first step, this limit WH may never be reached. Note that, in this case, it is expected to traverse the each image plane at least once. Thus the lower bound of the complexity is $\mathbf{O}(WH)$.

[0065] Seed selection may form a bottleneck in this extraction technique. To improve time efficiency, the algorithm may proceed in a multiscale fashion: only at the coarsest level is the full volume computed; at all subsequent levels, seeds are inherited from the previous level. To guarantee the existence of seeds at the coarsest level, the uniqueness condition that has been described in previous arrangements, is replaced by a winner-take-all strategy. That is, at each (u,v), we compute all voxels (u,v,d) where $d \in [-W_0/2, W_0/2]$ and choose the one that has the maximum correlation value.

[0066] Under this relaxed condition, some seeds may represent incorrect matches. To deal with this, we assign the seeds randomly to five different layers. As a result, five disparity maps are generated at the end of tracing. This allows

identifying and removing wrong matches. If no agreement can be reached, that point is left unmatched. At each level, extraction is performed for both the first and second images. Crosschecking is then conducted. Those pixels whose left and right disparities differ by more than one pixel are eliminated and recorded as unmatched. At the finest level, small holes are filled starting from the borders shows the final disparity map resulting from the improved algorithm. The execution time is reduced to about 1/6 of the previous version.

[0067] Assume the reduction rate between the two resolutions is 4 and the size of the correlation window is constant over all resolutions, the time complexity is reduced to $\mathbf{O}(WHS)$. Another merit of the multi-resolution version is that there is no need to prescribe a value for D.

[0068] The disparity map may then be manually edited at 135. This may allow the user to manually remove any information which appears out of place.

[0069] Shape inference is carried out at 140. The function of shape inference is to convert to the "dense" disparity map into a 3-D cluster of Euclidean points coordinates. Usually, the interest is in the shape appearance of the objects. Accordingly, this enables formation of a transformation to the final construction.

[0070] In the reconstruction stage, the correspondence information is transformed into 3-D Euclidean coordinates of the matched points. The operation carries out a two-step approach which includes projective reconstruction followed by Euclidean reconstruction.

[0071] The projective reconstruction may proceed by matrix factorization.

[0072] Kanade *et al.* has described a reconstruction algorithm using matrix factorization. The projections of n points may be considered in two views as $[u_{ij}, v_{ij}]^T$ where i=1,2 and j=1, ..., n. The following *measurement matrix* is defined:

$$W = \begin{bmatrix} \begin{bmatrix} u_{11} \\ v_{11} \end{bmatrix} & \cdots & \begin{bmatrix} u_{1n} \\ v_{1n} \end{bmatrix} \\ \begin{bmatrix} u_{21} \\ v_{21} \end{bmatrix} & \cdots & \begin{bmatrix} u_{2n} \\ v_{2n} \end{bmatrix} \end{bmatrix}.$$

[0073] The authors observed that, under orthographic or para-perspective projection, the aforementioned matrix is of rank 3. Then, a rank-3-factoriztion of the measurement matrix gives the affine reconstruction. One advantage of their algorithm is that all points are used concurrently and uniformly.

[0074] In applying the idea to perspective projection models, Chen and Medioni show that the following *modified measurement matrix* is of rank 4:

$$W = \begin{bmatrix} \begin{bmatrix} u'_{11} \\ v'_{11} \\ 1 \end{bmatrix} & \cdots & \begin{bmatrix} u'_{1n} \\ v'_{1n} \\ 1 \end{bmatrix} \\ \begin{bmatrix} u'_{21} \\ v'_{21} \\ 1 \end{bmatrix} & \cdots & \begin{bmatrix} u'_{2n} \\ v'_{2n} \\ 1 \end{bmatrix} \end{bmatrix}$$

[0075] where each column denotes a pair of corresponding pixels after rectification. Thus a rank-4-factorization produces a projective reconstruction (section):

$$W = P_{6\times 4} \cdot Q_{4\times n} = \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} \begin{bmatrix} Q_1 & \cdots & Q_n \end{bmatrix}, \tag{3}$$

[0076] where P_1 and P_2 are the 3×4 matrices of the two cameras, and Q_i's are the homogeneous coordinates of the points. Such a factorization may be carried out using Singular Value Decomposition (SVD).

[0077] Next, the so-far obtained projective reconstruction is converted into the first canonical form which is a prerequisite of our Euclidean reconstruction algorithm.

[0078] Let $P_1 = [P_{11} \quad \boldsymbol{p}_1]$. It is known that $C_1 = -P_{11}^{-1}\boldsymbol{p}_1$ is the first projection center. The stereo rig can be translated so that C_1 is coincident with the world origin. Let the translation matrix be

$$B = \begin{bmatrix} I & -P_{11}^{-1}\boldsymbol{p}_1 \\ 0 & 1 \end{bmatrix},$$

then

$$P_1 B = [P_{11} \quad \boldsymbol{p}_1] \begin{bmatrix} I & -P_{11}^{-1}\boldsymbol{p}_1 \\ 0 & 1 \end{bmatrix} = [P_{11} \quad 0].$$

Thus

$$W = \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} BB^{-1} \begin{bmatrix} Q_1 & \cdots & Q_n \end{bmatrix} = \begin{bmatrix} P_{11} & 0 \\ P_{21} & \boldsymbol{p}_2 \end{bmatrix} \begin{bmatrix} Q_1' & \cdots & Q_n' \end{bmatrix} \quad (4)$$

is the desired canonical form for stereo projective reconstruction.

[0079] Now that the world coordinate system (the origin and the axes) is coincident with that of the first camera, Euclidean reconstruction is equivalent to finding the Projective Distortion Matrix H such that

$$\begin{bmatrix} P_{11} & 0 \end{bmatrix} H = \begin{bmatrix} A_1 & 0 \end{bmatrix} I, \quad (5a)$$

and

$$\begin{bmatrix} P_{21} & \boldsymbol{p}_2 \end{bmatrix} H = \mu \begin{bmatrix} A_2 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \quad (5b)$$

where μ compensates for the relative scaling between the two equations and A1 and A2 are diagonal matrices consisting focal length of the two cameras: $A_1 = \begin{bmatrix} f_1 & 0 & 0 \\ 0 & f_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $A_2 = \begin{bmatrix} f_2 & 0 & 0 \\ 0 & f_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. Since H is defined up to a scale factor, we set one of its elements to be 1:

$$H = \begin{bmatrix} H_1 & h_1 \\ h^T & 1 \end{bmatrix}.$$

Then, (5a) becomes,

$$\begin{bmatrix} P_{11} & 0 \end{bmatrix} \begin{bmatrix} H_1 & \boldsymbol{h}_1 \\ \boldsymbol{h}^T & 1 \end{bmatrix} = \begin{bmatrix} P_{11}H_1 & P_{11}\boldsymbol{h}_1 \end{bmatrix} = \begin{bmatrix} A_1 & 0 \end{bmatrix}$$

which implies $\boldsymbol{h}_1$=0 and $H_1 = P_{11}^{-1}A_1$. Thus

$$H = \begin{bmatrix} P_{11}^{-1}A_1 & 0 \\ \boldsymbol{h}^T & 1 \end{bmatrix}. \tag{6}$$

Plug (6) into (5b),

$$\begin{bmatrix} P_{21} & \boldsymbol{p}_2 \end{bmatrix} \begin{bmatrix} P_{11}^{-1}A_1 & 0 \\ \boldsymbol{h}^T & 1 \end{bmatrix} = \begin{bmatrix} P_{21}P_{11}^{-1}A_2 + \boldsymbol{p}_2\boldsymbol{h}^T & \boldsymbol{p}_2 \end{bmatrix} = \mu \begin{bmatrix} A_2R & A_2T \end{bmatrix}$$

which generates

$$P_{21}P_{11}^{-1}A_1 + \boldsymbol{p}_2\boldsymbol{h}^T = \mu A_2 R = \mu \begin{bmatrix} f_2R_1 \\ f_2R_2 \\ R_3 \end{bmatrix} = \begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix} \tag{7a}$$

and

$$\boldsymbol{p}_2 = \mu A_2 T\,. \tag{7b}$$

Since R is a rotation matrix, (7a) further expands into the following 5 constraints on f_1, f_2, and $\boldsymbol{h}$:

$$M_1 \bullet M_2 = M_2 \bullet M_3 = M_3 \bullet M_1 = 0\,, \tag{8a}$$

$$\|M_1\| = \|M_2\| = f_2\|M_3\|\,, \tag{8b}$$

Once f_1, f_2, and $\boldsymbol{h}$ are computed, H can be obtained from (6). R, T and μ are obtained from (4.7). To determine the initial value for

H, let $R \approx I$, $\mu \approx 1$, and $A_1 \approx A_2 = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} = A$, since the two cameras have similar orientation and focal length. It follows that

$$H_1 = P_{11}^{-1} A \quad \text{and} \quad \boldsymbol{p}_2 \boldsymbol{h}^T = (I - P_{21} P_{11}^{-1}) A .$$

[0080] Thus, an approximate Euclidean reconstruction can be achieved solely depending on f. We have developed an interactive tool to let the user input f, and adjust its value until the approximation looks reasonable.

[0081] Initial work on this invention has carried out its operation on faces. Faces may be difficult to reconstruct due to their smooth shape, and relative lack of prominent features. Moreover, faces may have many applications including teleconferencing and animation.

[0082] Other embodiments are within the disclosed invention.